AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method in a computer system having a global descriptor table register for executing code during a system management mode interrupt (SMI), the method comprising:

upon occurrence of ~~an~~ the SMI,

saving state of the computer system;

switching the computer system to protected mode;

replacing first contents of the global descriptor table register that point to a first global descriptor table in use when the system management mode interrupt occurred with second contents that point to a second global descriptor table that is distinct from the first global descriptor table;

executing 32-bit code using the second global descriptor table; and

upon completion of the execution of the 32-bit code,

restoring the saved state of the computer system; and

returning from the occurrence of the SMI.

2. (Previously Presented) The method of claim 1 wherein the 32-bit code is an operating system kernel for loading and running programs during the occurrence of the system management mode interrupt.

3. (Previously Presented) The method of claim 2 wherein the programs are Windows Portable Executable programs.

4. (Previously Presented) The method of claim 1 wherein the computer system is based on an Intel Pentium processor.

5. (Currently Amended) A method in a computer system having a global descriptor table register for executing code during an SMI, the method comprising:

upon occurrence of ~~an~~ the SMI,

switching the computer system from the system management mode to a second mode;

replacing first contents of the global descriptor table register that point to a first global descriptor table in use when the system management mode interrupt occurred with second contents that point to a second global descriptor table that is distinct from the first global descriptor table; and

executing code using the second global descriptor table.

6. (Previously Presented) The method of claim 5 including upon completion of the execution of the code, returning from the occurrence of the SMI.

7. (Previously Presented) The method of claim 5 including:

saving state of the computer system; and

upon completion of the execution of the code,

restoring the saved state of the computer system; and

returning from the occurrence of the SMI.

8. (Previously Presented) The method of claim 5 wherein the executing executes 32-bit code.

9. (Previously Presented) The method of claim 5 wherein the code is an operating system kernel for loading and running programs during the occurrence of the SMI.

10. (Previously Presented) The method of claim 9 wherein the programs are portable executables.

11. (Previously Presented) The method of claim 5 wherein the computer system is based on an Intel processor.

12. (Previously Presented) The method of claim 5 wherein the computer system is based on an Intel-compatible processor.

13. (Previously Presented) The method of claim 5 wherein the executed code is selected from the group consisting of a remote console program, a remote boot program, a remote diagnostics program, a remote restart program, and a debugging program.

14. (Previously Presented) The method of claim 5 wherein the second mode is protected mode.

15. (Previously Presented) The method of claim 5 wherein the computer system has a foreground operating system, and wherein the code executes transparently to the foreground operating system.

16. (Previously Presented) The method of claim 5 wherein the computer system has a foreground operating system, and wherein the code executes even if the foreground operating system has crashed or stopped.

17. (Previously Presented) The method of claim 5 wherein the computer system has a foreground operating system, and wherein the code executes when the foreground operating system crashes or stops.

18. (Currently Amended) A computer-readable medium containing instructions for an SMI routine that allows execution of ~~code~~ a Portable Executable program that resides in physical address space above address 0x100000, by a method performed in response to receipt of the SMI comprising:

saving state of the processor;

switching the processor to protected mode; and

before returning from the SMI, executing ~~code~~ the Portable Executable program that resides in physical address space above address 0x100000.

19. (Currently Amended) The computer-readable medium of claim 18 including after executing the ~~code~~ Portable Executable program, restoring the saved state of the processor and returning from the SMI.

20. (Currently Amended) The computer-readable medium of claim 18 wherein the ~~code~~ Portable Executable program is executed using a global descriptor table that is different from the global descriptor table in use when the SMI occurred.

21. (Previously Presented) The computer-readable medium of claim 18 wherein the processor is a Pentium-based processor.

22. (Currently Amended) The computer-readable medium of claim 18 wherein the ~~executed code~~ Portable Executable program is 32-bit flat address space code.

23. (Previously Presented) The computer-readable medium of claim 18 wherein the instructions are loaded into system management memory by a BIOS.

24. (Currently Amended) The computer-readable medium of claim 18 wherein the ~~code~~ Portable Executable program is loaded into memory from a ROM.

25. (Currently Amended) The computer-readable medium of claim 18 wherein the ~~code~~ Portable Executable program is loaded into memory from a Flash ROM.

26. (Currently Amended) The method of claim 1 wherein a processor switches to system management mode and executes ~~an~~ the SMI in response to a signal received on ~~a package pin~~ an input line of the processor.

27. (Currently Amended) The method of claim 26 wherein the ~~package pin~~ input line is ~~the SMI# package pin~~ an SMI input line.

28. (Canceled)

29. (Currently Amended) The method for claim 1 wherein a processor switches to system management mode and executes ~~an~~ the SMI in response to a message received via a front side bus of the processor.

30. (Previously Presented) The method of claim 1 wherein a processor chip set is the source of the SMI.

31. (Previously Presented) The method of claim 1 wherein a Northbridge controller is the source of the SMI.

32. (Previously Presented) The method of claim 1 wherein a Southbridge controller is the source of the SMI.

33. (Previously Presented) The method of claim 1 wherein an electronic circuit is the source of the SMI.

34. (Previously Presented) The method of claim 1 wherein returning from the occurrence of the interrupt is accomplished by executing an RSM instruction.

35. (Canceled)

36. (Currently Amended) The method of claim 5 wherein a processor switches to system management mode and executes ~~an~~ the SMI in response to a signal received on ~~a package pin~~ an input line of the processor.

37. (Currently Amended) The method of claim 36 wherein the ~~package pin~~ input line is ~~the SMI# package pin~~ an SMI input line.

38. (Canceled)

39. (Currently Amended) The method of claim 5 wherein a processor switches to system management mode and executes ~~an~~ the SMI in response to a message received via a front side bus of the processor.

40. (Previously Presented) The method of claim 5 wherein a processor chip set is the source of the SMI.

41. (Previously Presented) The method of claim 5 wherein a Northbridge controller is the source of the SMI.

42. (Previously Presented) The method of claim 5 wherein a Southbridge controller is the source of the SMI.

43. (Previously Presented) The method of claim 5 wherein an electronic circuit is the source of the SMI.

44. (Previously Presented) The method of claim 5, further comprising returning from the occurrence of the interrupt by executing an RSM instruction.

45. (Canceled)

46. (Canceled)

47. (Currently Amended) The computer-readable medium of ~~claim 46~~ claim 18 wherein a processor switches to system management mode and executes ~~an~~ the SMI in response to a signal received on ~~a package pin~~ an input line of the processor.

48. (Currently Amended) The computer-readable medium of claim 47 wherein the ~~package pin~~ input line is ~~the SMI# package pin~~ an SMI input line.

49. (Canceled)

50. (Currently Amended) The computer-readable medium of ~~claim 46~~ claim 18 wherein a processor switches to system management mode and executes ~~an~~ the SMI in response to a message received via a front side bus of the processor.

51. (Currently Amended) The computer-readable medium of ~~claim 46~~ claim 18 wherein a processor chip set is the source of the SMI.

52. (Currently Amended) The computer-readable medium of ~~claim 46~~ claim 18 wherein a Northbridge controller is the source of the SMI.

53. (Currently Amended) The computer-readable medium of ~~claim 46~~ claim 18 wherein a Southbridge controller is the source of the SMI.

54. (Currently Amended) The computer-readable medium of ~~claim 46~~ claim 18 wherein an electronic circuit is the source of the SMI.

55. (Previously Presented) The computer-readable medium of claim 18, the method further comprising returning from the interrupt by executing an RSM instruction.

56. (Canceled)